

AMENDMENTS TO THE CLAIMS

Claims 1-24 were originally pending. Please amend claims 1-6, 8-21, and 23. Please cancel claims 7, 22, and 24.

The following listing of claims replaces all prior versions, and listings of claims in the application.

Listing of Claims:

1. (Currently amended) A method to be implemented in a computer system comprising a processor and a memory, the method for managing a run queue comprising a first plurality of threads sorted with respect to one another based on thread priority, the method comprising:

in a deterministic amount of time equivalent to an amount of time to insert a single thread into the run queue, associating a second plurality of threads that is priority sorted with the run queue in a manner that maintains a priority based scheduling semantic of the run queue.

2. (Currently amended) A method as recited in claim 1, wherein the second plurality of threads comprises a root thread, and wherein associating the second plurality of threads with the run queue further comprises: inserting only the root thread into the run queue.

1 3. (Currently amended) A method as recited in claim 1, wherein the
2 associating the second plurality of threads with the run queue further comprises:
3 inserting each thread in the second plurality of threads into the run queue
4 independent of any additional other queue access.
5

6 4. (Currently amended) A method as recited in claim 1, wherein
7 associating the second plurality of threads with the run queue further comprises:
8 inserting only a root thread of the second plurality of threads into the run queue ~~to~~
9 ~~represent the second plurality of nodes.~~
10

11 5. (Currently amended) A method as recited in claim 1, wherein
12 associating the second plurality of threads with the run queue further comprises:

13 inserting only a root thread of the second plurality of threads into the run
14 queue; and

15 wherein the method further comprises:

16 removing the root thread from the run queue; and

17 responsive to removing the root thread, inserting a next thread of the
18 second plurality of threads into the run queue such that the priority based
19 scheduling semantic of the run queue is preserved.
20
21
22
23
24
25

1 6. (Currently amended) A method as recited in claim 1, wherein the
2 method further comprises:

3 inserting a root thread of the second plurality of threads into the run queue;
4 removing the root thread from the run queue for execution; and
5 responsive to removing the root thread and independent of any additional
6 other queue access, inserting a next thread of the second plurality of threads into
7 the run queue.

8
9 7. (Canceled).

10
11 8. (Currently amended) A system for managing a run queue, the run
12 queue comprising a first plurality of threads, each thread in the first plurality of
13 threads having a respective priority, the first plurality of threads being sorted such
14 that a thread having a high priority is removed from the run queue before a thread
15 having a lower priority, the system comprising:

16 a memory for storing the run queue and computer-executable instructions;
17 a processor operatively coupled to the memory, the processor being
18 configured to execute the computer-executable instructions for:

19 in a ~~deterministic~~ an amount of time to insert a single thread into the
20 run queue, associating the second plurality of threads that is priority sorted with
21 the run queue, the associating maintaining a priority based scheduling semantic of
22 the run queue.

1 9. (Currently amended) A system as recited in claim 8, wherein
2 associating the second plurality of threads with the run queue is performed
3 independent of more than a single other queue access.
4

5 10. (Currently amended) A system as recited in claim 8, wherein the
6 second plurality of threads comprises a root thread operatively coupled to one or
7 more other threads of the second plurality of threads, each of the one or more other
8 threads having a respective priority that is a lower priority or an equal priority as
9 compared to a priority of the root ~~node~~ thread.
10

11 11. (Currently amended) A system as recited in claim 8, wherein
12 associating the second plurality of threads with the run queue further comprises:
13 inserting only a root thread of the second plurality of threads into the run queue.
14

15 12. (Currently amended) A system as recited in claim 8, wherein
16 associating the second plurality of threads with the run queue further comprises:
17 inserting only a root thread of the second plurality of threads into the run queue ~~to~~
18 ~~represent the second plurality of threads.~~
19
20
21
22
23
24
25

1 13. (Currently amended) A system as recited in claim 8:
2 wherein the first plurality of threads is a first linked list data structure;
3 wherein the second plurality of threads is a second linked list data structure
4 comprising a root node that is operatively coupled to one or more other threads in
5 the second plurality of threads; and
6 wherein the single insert operation is an operation comprising inserting the
7 root node into a position in the first linked list data structure.

8
9 14. (Currently amended) A system as recited in claim 8, wherein
10 associating the second plurality of threads with the run queue further comprises:
11 inserting only a root thread of the second plurality of threads into the run
12 queue; and
13 wherein the method further comprises:
14 removing the root thread from the run queue; and
15 responsive to removing the root thread, inserting a next thread of the
16 second plurality of threads into the run queue such that a priority based scheduling
17 semantic of the run queue is preserved.

18
19 15. (Currently amended) A system as recited in claim 8, wherein the
20 processor is further configured to execute computer program instructions for:
21 inserting a root thread of the second plurality of threads into the run queue;
22 removing the root thread from the run queue for execution; and
23 responsive to removing the root thread and independent of any additional
24 other queue access, inserting a next thread of the second plurality of threads into
25 the run queue.

1
2 16. (Currently amended) A computer-readable storage medium
3 comprising computer-executable instructions to manage a run queue sorted with to
4 one another based on thread priority, the computer-executable instructions
5 comprising instructions for:

6 in a deterministic amount of time that is independent of the number of
7 threads in a second plurality of threads that is priority sorted, the deterministic
8 amount of time being a time to insert a single thread into the run queue,
9 associating the second plurality of threads with the run queue in a manner that
10 maintains a priority based scheduling semantic of the run queue.

11
12 17. (Currently amended) A computer-readable storage medium as
13 recited in claim 16, wherein the second plurality of threads comprises a root thread
14 that is operatively coupled to one or more other threads of the second plurality of
15 threads, and wherein the instructions for associating further comprise:

16 inserting only the root thread into the first plurality of threads ~~to represent~~
17 ~~the second plurality of threads.~~

18
19 18. (Currently amended) A computer-readable storage medium as
20 recited in claim 16, wherein the first plurality of threads is a first linked list data
21 structure, the second plurality of threads is a second linked list data structure
22 comprising a root node that is operatively coupled to one or more other threads in
23 the second plurality of threads, and the deterministic amount of time is a result of a
24 single insert operation to insert the root node into the first linked list data structure.
25

1 19. (Currently amended) A computer-readable storage medium as
2 recited in claim 16, wherein the instructions for associating further comprise:

3 inserting only a root thread of the second plurality of threads into the first
4 plurality of threads;

5 and wherein the computer-executable instructions further comprise
6 instructions for:

7 removing the root thread from the run queue; and

8 responsive to removing the root thread, inserting a next thread of the
9 second plurality of threads into the first plurality in a manner that maintains a
10 priority based scheduling semantic of the run queue .

11
12 20. (Currently amended) A computer-readable storage medium as
13 recited in claim 19, wherein the ~~aets~~ instructions for inserting the next thread are
14 performed independent of an other queue.

15
16 21. (Currently amended) A computer-readable storage medium as
17 recited in claim 16, wherein the instructions for associating further comprise
18 instructions for:

19 inserting a root thread of the second plurality of threads into the first
20 plurality;

21 removing the root thread from the first plurality for execution; and

22 responsive to removing the root thread, inserting a next thread of the
23 second plurality of threads into the first plurality of threads independent of any
24 additional access to another different queue.

1 22. (Canceled).

2
3 23. (Currently amended) A computer-readable medium comprising a
4 run queue data structure, the run queue data structure comprising:

5 a first dimension data field comprising a first plurality of threads sorted
6 with respect to thread priority; and

7 a second dimension data field comprising a second plurality of threads
8 sorted based on thread priority, the second plurality of threads comprising a root
9 thread and one or more other threads.

10
11 24. (Canceled).